



# Certificación PCAP

Lección 6: Módulos y paquetes, excepciones, generadores y cierres y procesamiento de ficheros

# Actividad 1

## ACTIVIDAD LECCIÓN 6

### Objetivos

- | Familiarizarse con los conceptos vistos en esta lección

### Contenido correspondiente a lección 6:

- Excepciones
- Funciones Lambda
- Procesando archivos

### Actividad relacionada con la lección 5:

El objetivo de esta actividad es crear un pequeño código para leer las notas de los alumnos desde un fichero de texto llamado 'fichero\_texto.txt'. El fichero de texto contiene una nota por línea, pero puede haber líneas por medio vacías o con caracteres que no deben contemplarse. Al terminar de leer el fichero, debe mostrar la lista con las notas leídas, las notas que tiene con sobresaliente (aquellas con un 9.0 o más) y mostrar el número de líneas excluidas ya que no corresponden a notas (líneas con caracteres o vacías). Al mostrar las notas éstas deben estar ordenadas de menos a más.

Por ejemplo, para el fichero de texto:

```
1 9.5
2 10
3 a
4 9.5
5 4
6
7 6.4
8 blanco
9 9
10 8
```

Deberá mostrar:

```
Las notas leídas son: [4.0, 6.4, 8.0, 9.0, 9.5, 9.5, 10.0]
Notas con sobresaliente: [9.0, 9.5, 9.5, 10.0]
Líneas excluidas: 3
```

# Actividad 2

## ACTIVIDAD LECCIÓN 6

### Objetivos

- | Profundizar en los conceptos vistos en esta lección

### Contenido correspondiente a lección 6:

- Módulos y Paquetes
- Excepciones
- Generadores y cierres
- Funciones Lambda
- Procesando archivos

### Actividad relacionada con la lección 6:

Las preguntas sobre los temas vistos en esta lección son muy comunes en el examen de certificación PCAP, por lo que con esta actividad se pretende trabajar en los diferentes conceptos vistos en esta lección. En esta actividad debes enviar para cada pregunta la opción u opciones correctas, así como una breve explicación de tu elección. Es importante que no te olvides de enviar la breve explicación de tu elección ya que si no la envías no se te evaluará la pregunta.

### Pregunta 1:

Si tengo dos módulos definidos como:

```
# mimodulo1.py
print("MODULO1", end=" ")
MODULO = 1
```

```
# mimodulo2.py
print("MODULO2", end=" ")
MODULO = 2
|
```

Si importo ambos módulos del siguiente modo, ¿cuál de las siguientes afirmaciones es correcta? (1 opción)

```
from mimodulo1 import *  
from mimodulo2 import *  
from mimodulo1 import *  
from mimodulo2 import *  
print(MODULO)
```

- A) Al ejecutarse el código mostrará  
MODULO1 MODULO2 1
- B) Al ejecutarse el código mostrará  
MODULO1 MODULO2 2
- C) Al ejecutarse el código mostrará  
MODULO2 MODULO1 1
- D) Al ejecutarse el código mostrará  
MODULO2 MODULO1 2
- E) Al ejecutarse el código mostrará  
MODULO1 MODULO2 MODULO1 MODULO2 1
- F) Al ejecutarse el código mostrará  
MODULO1 MODULO2 MODULO1 MODULO2 2
- G) Al ejecutarse el código mostrará  
1
- H) Al ejecutarse el código mostrará  
2
- I) Al ejecutarse dará un error.

### Pregunta 2:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
try:  
    mi_lista = [1, 2 , 3]  
    print(mi_lista[3])  
except LookupError:  
    print("Error: LookupError")  
except IndexError:  
    print("Error: IndexError")  
except:  
    print("Error general")
```

- A) 3
- B) Error: LookupError
- C) Error: IndexError
- D) Error general

**Pregunta 3:**

¿Cuál o cuáles de las siguientes sentencias aparecerán en la salida por pantalla al ejecutar el siguiente código? (1 o varias opciones)

```
try:
    x = int("3,4")
    print("Todo OK")
except ValueError:
    print("Error: ValueError")
except:
    print("Error general")
else:
    print("ELSE")
finally:
    print("FINALLY")
```

- A) Todo OK
- B) Error: ValueError
- C) Error general
- D) ELSE
- E) FINALLY

**Pregunta 4:**

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
def o(p):  
    def q():  
        return p * '*'  
    return q  
  
r = o(1)  
s = o(2)  
print(r() + s())
```

- A) \*\*
- B) \*\*\*
- C) \*\*\*\*
- D) \*
- E) Devolverá un error

**Pregunta 5:**

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
def I():  
    s = '12345678'  
    for c in s[1::2]:  
        yield int(c)  
  
mi_lista = [i for i in I()]  
mi_lista2 = list(map(lambda x: x ** 2, mi_lista))  
print(mi_lista2)
```

- A) []
- B) [1, 2, 3, 4, 5, 6, 7, 8]
- C) [1, 4, 9, 16, 25, 36, 49, 64]
- D) [2, 4, 6, 8]
- E) [4, 16, 36, 64]
- F) [1, 3, 5, 7]
- G) [1, 9, 25, 49]
- H) Devolverá un error